

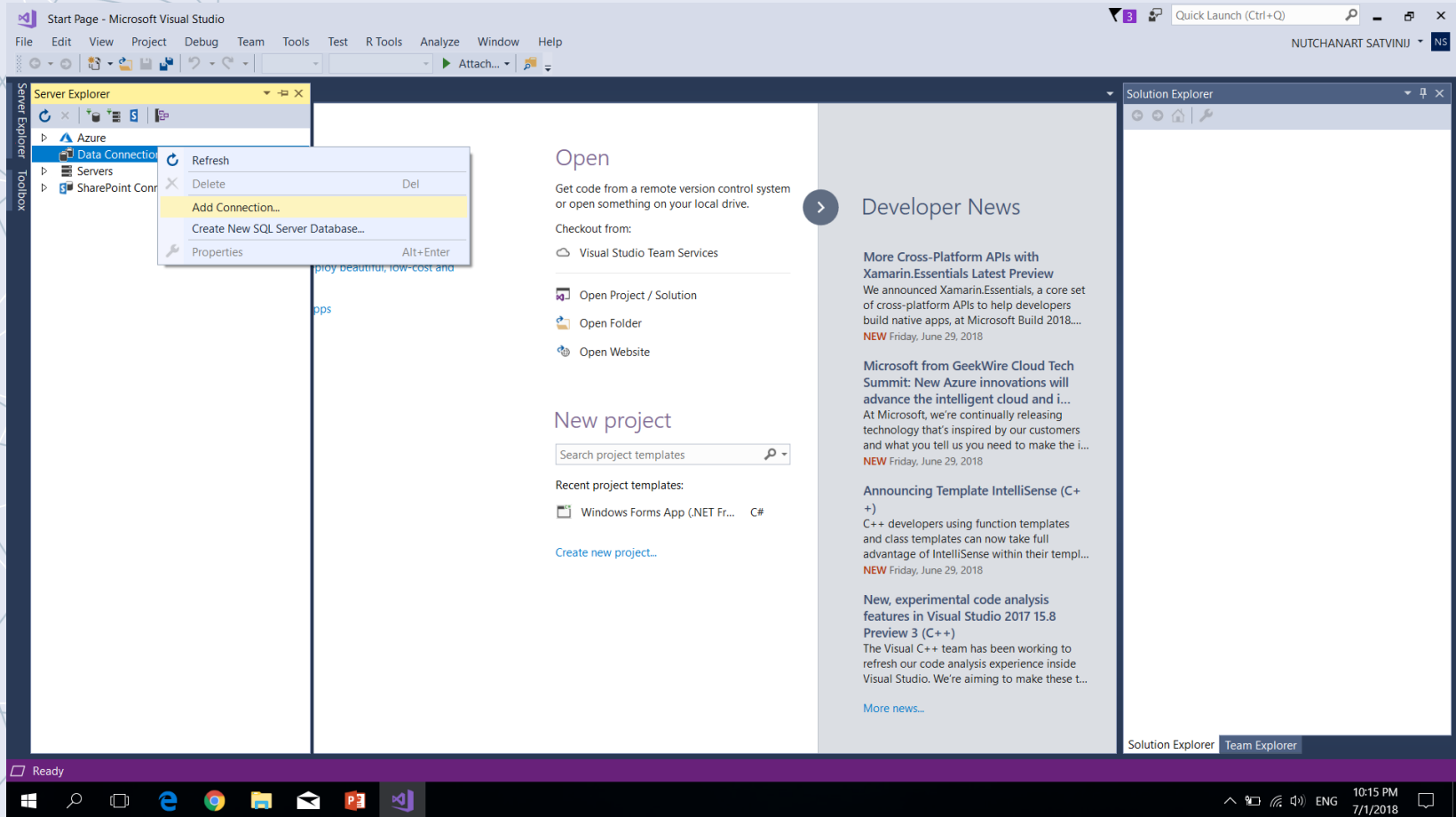
ระบบฐานข้อมูล

Chapter 02 : Basic SQL

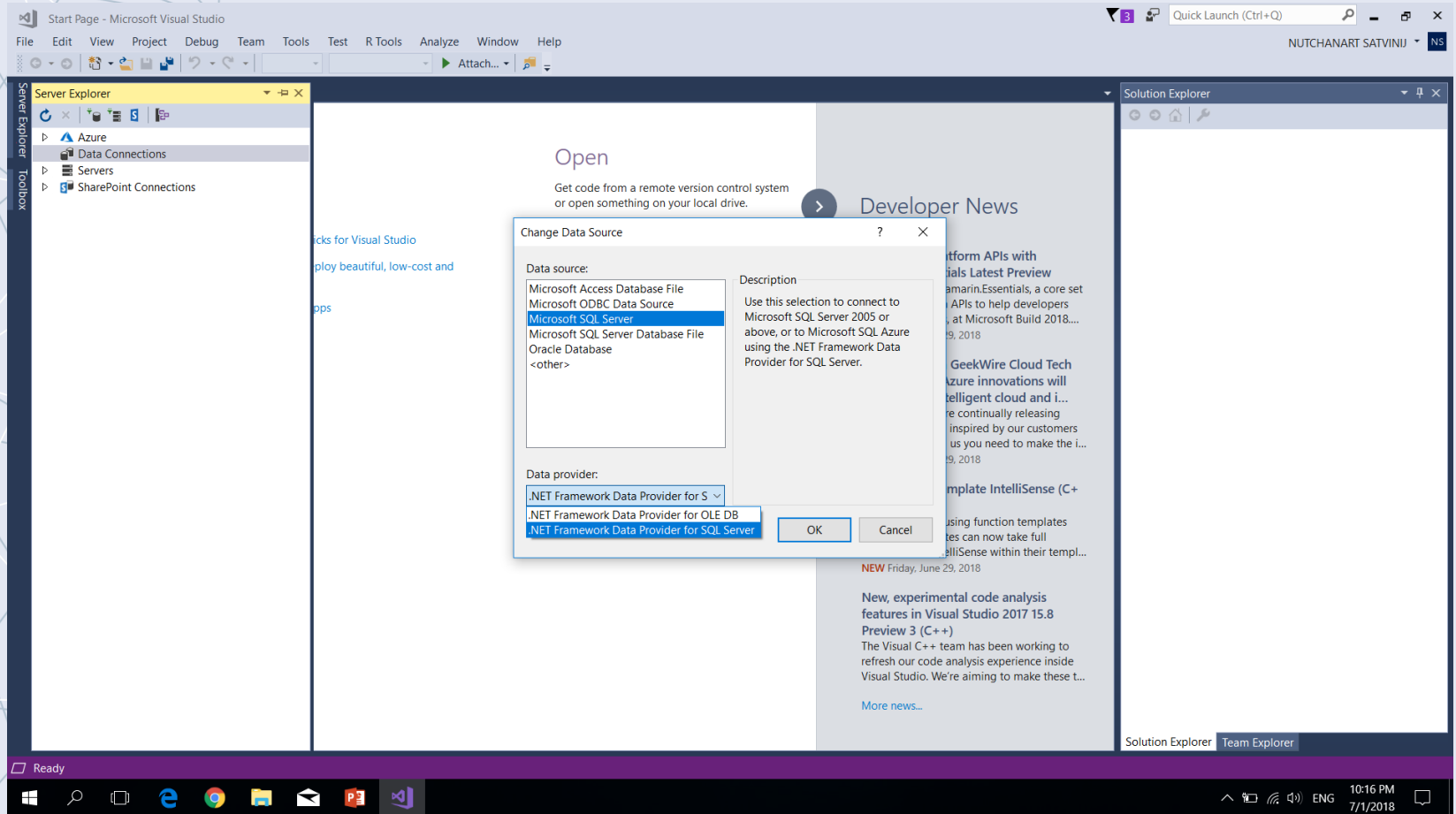
อ.นุชนาถ สัตย์วินิจ

ภาควิชาวิศวกรรมคอมพิวเตอร์ วิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์

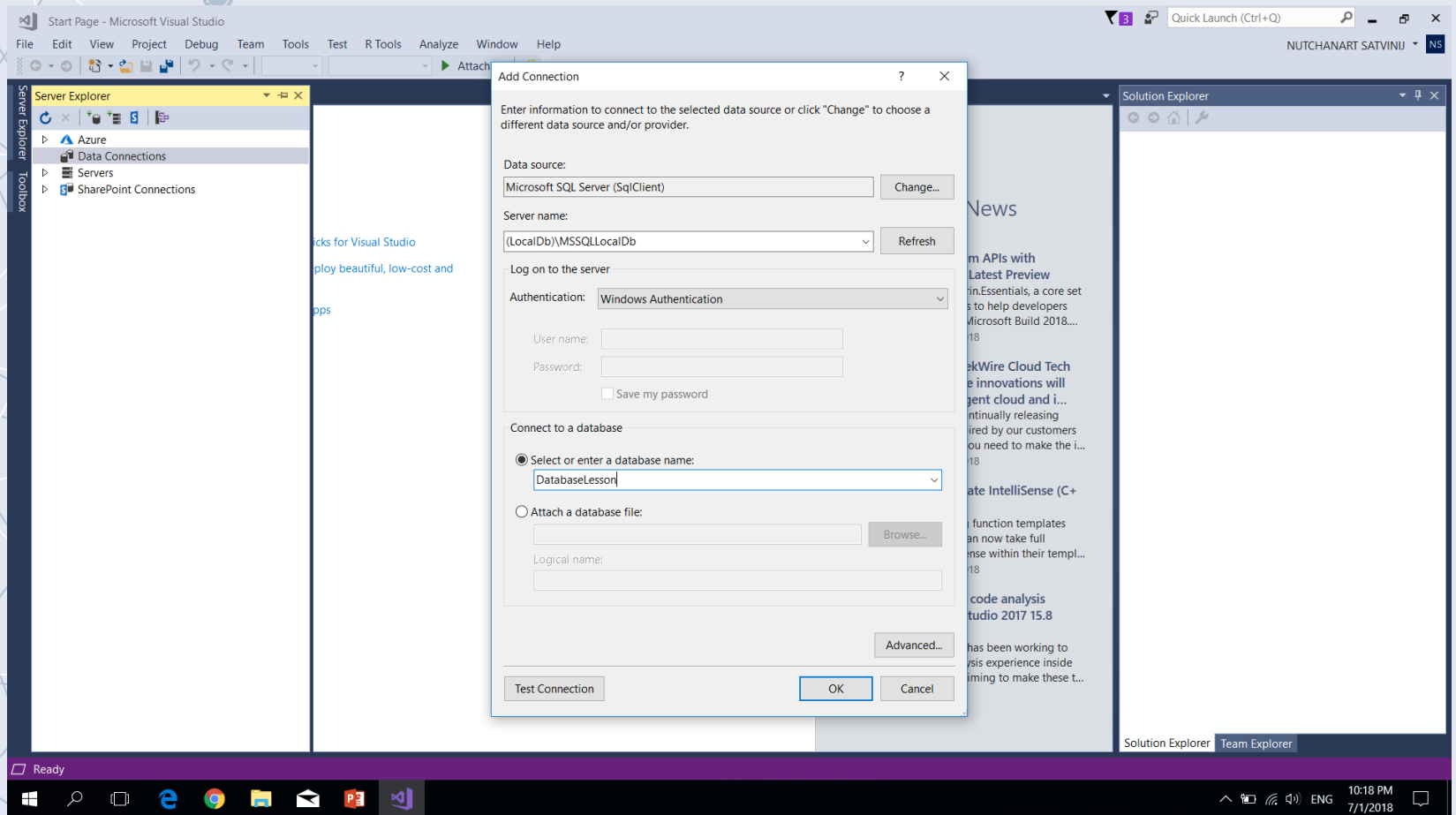
Getting Started



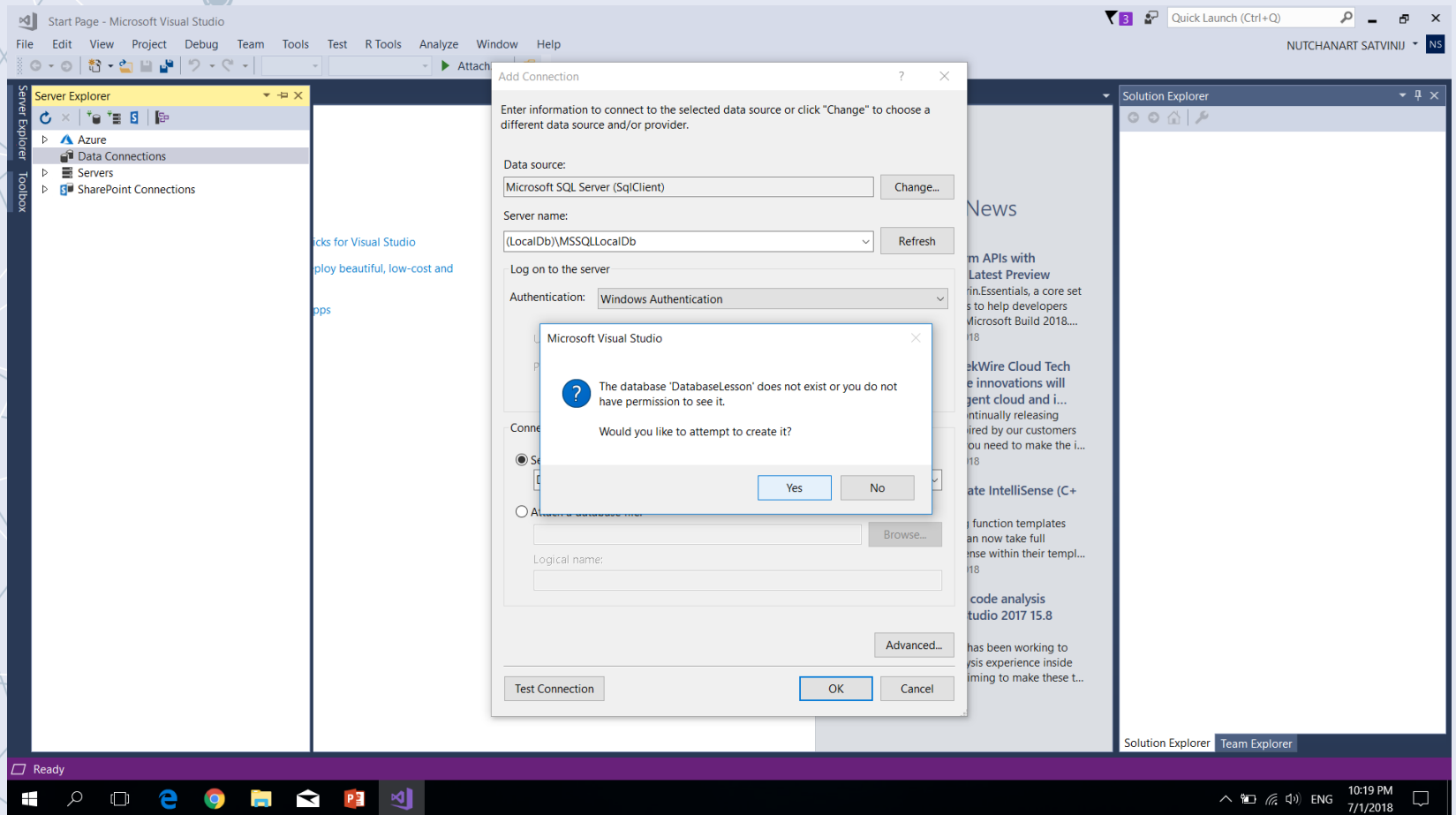
Getting Started



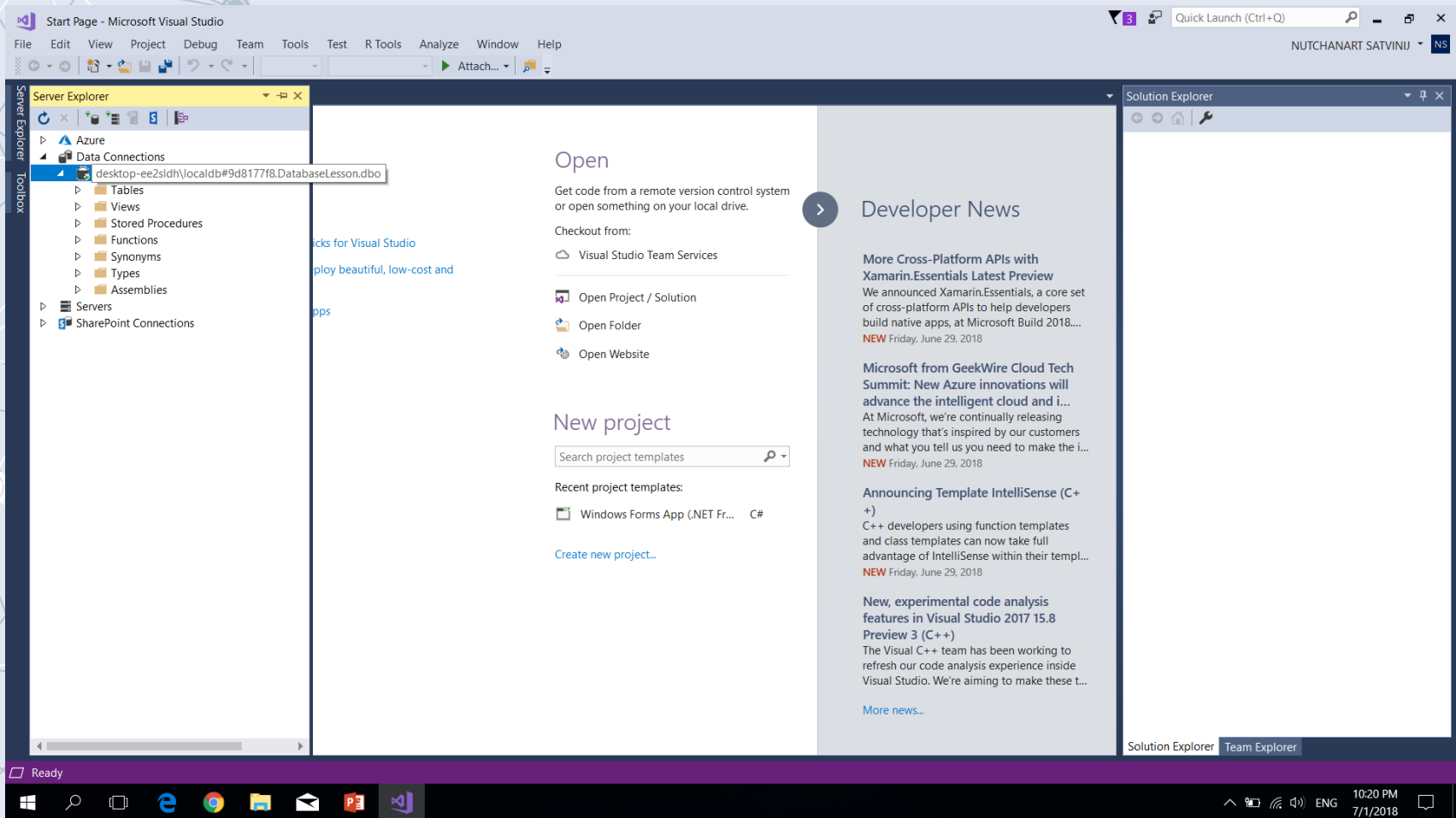
Getting Started



Getting Started



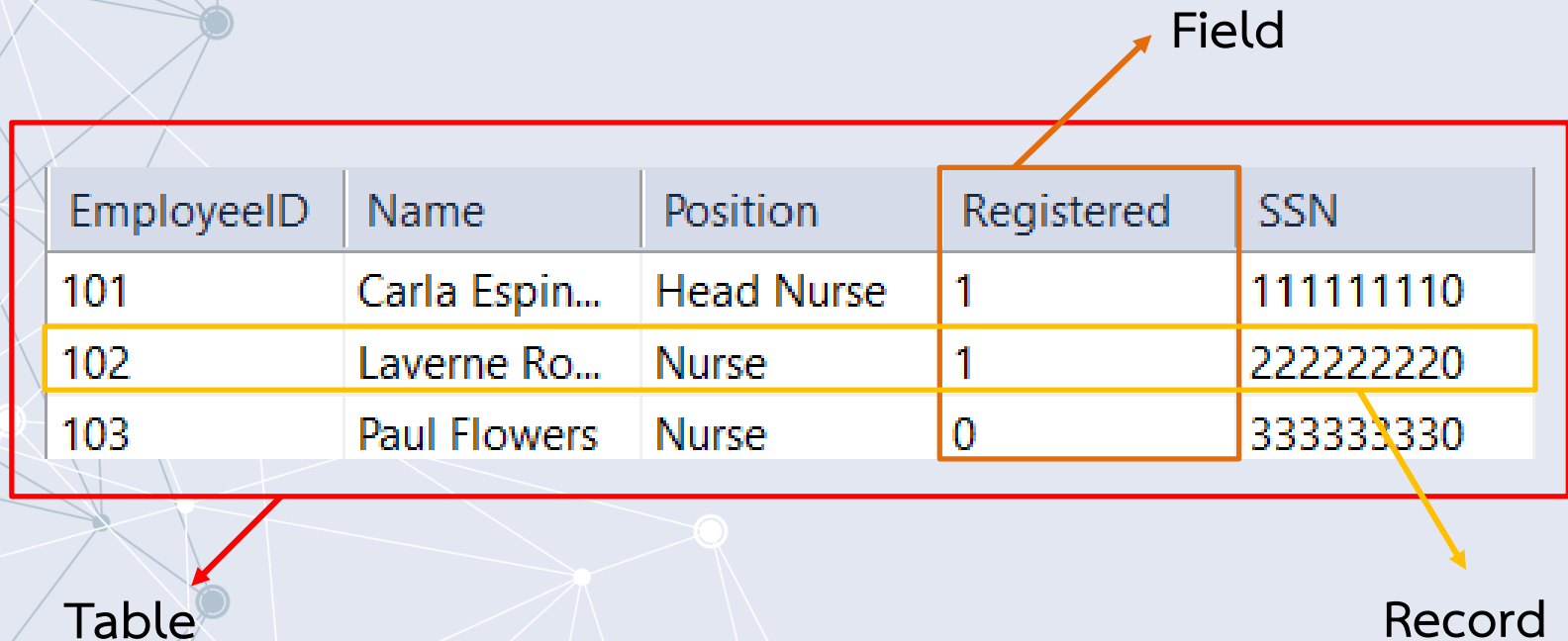
Getting Started



What is SQL

- SQL = Structured Query Language
- SQL = Computer language for accessing and manipulating database systems
- SQL is used for managing data in relational database management system

Table in Relational Database System

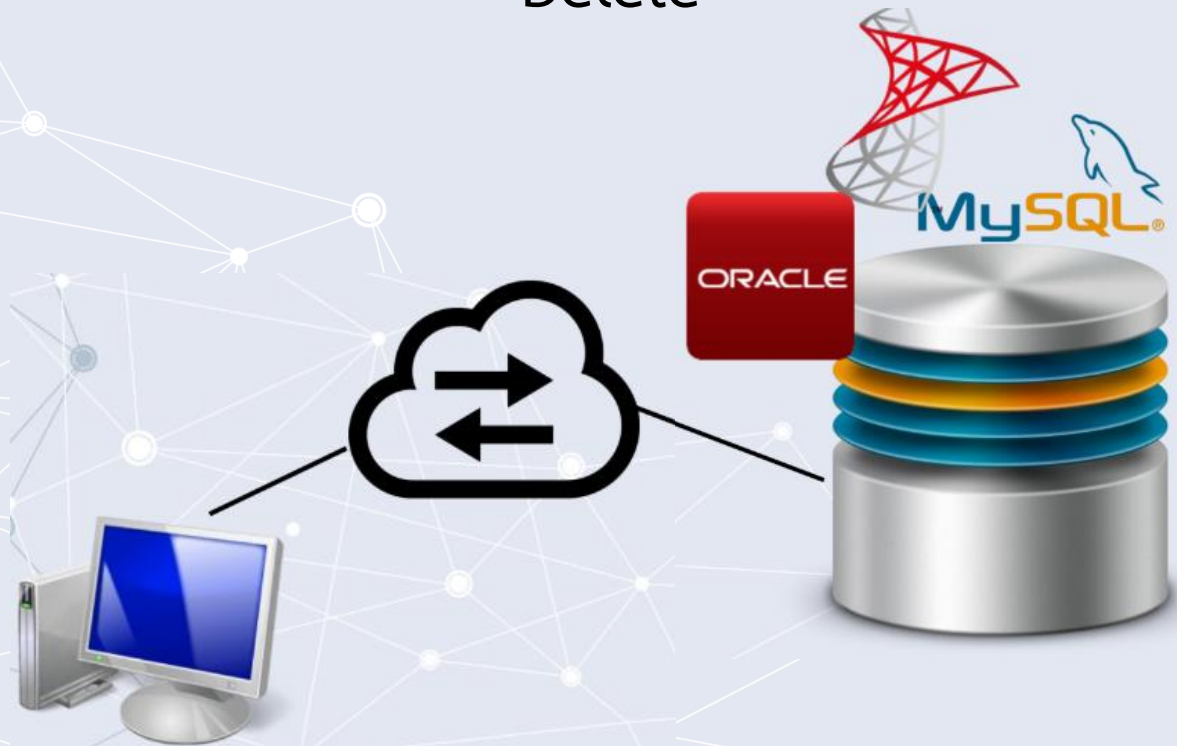


The diagram illustrates a table in a relational database system. A red rectangular border encloses the entire table structure. An orange arrow points from the word 'Field' to the 'Registered' column header. A yellow arrow points from the word 'Record' to the third row of data. A red arrow points from the word 'Table' to the bottom-left corner of the table's border.

EmployeeID	Name	Position	Registered	SSN
101	Carla Espin...	Head Nurse	1	111111110
102	Laverne Ro...	Nurse	1	222222220
103	Paul Flowers	Nurse	0	333333330

Interacting Database

- Define
- Create
- Select
- Insert
- Update
- Delete



Some of The Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database

Some of The Most Important SQL Commands

- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

CREATE table

```
CREATE TABLE table_name
```

```
(  
  column_name1 data_type(size) constraint_name,  
  column_name2 data_type(size) constraint_name,  
  column_name3 data_type(size) constraint_name,  
  ...  
)
```

CREATE table

- **constraint_name**
 - NOT NULL - Indicates that a column cannot store NULL value
 - UNIQUE - Ensures that each row for a column must have a unique value
 - PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Ensures that a column (or combination of two or more columns) have a unique identity which helps to find a particular record in a table more easily and quickly
 - FOREIGN KEY - Ensure the referential integrity of the data in one table to match values in another table
 - CHECK - Ensures that the value in a column meets a specific condition
 - DEFAULT - Specifies a default value for a column

Example CREATE table

```
CREATE TABLE Persons  
(  
    P_Id int NOT NULL PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
)
```

Example CREATE table

```
CREATE TABLE Persons
```

```
(  
    P_Id int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255),  
    CONSTRAINT pk_PersonID PRIMARY KEY (P_Id,LastName)  
)
```

Example CREATE table

```
CREATE TABLE Orders  
(  
    O_Id int NOT NULL PRIMARY KEY,  
    OrderNo int NOT NULL,  
    P_Id int FOREIGN KEY REFERENCES Persons(P_Id)  
)
```


Example CREATE table

```
CREATE TABLE Persons
```

```
(
```

```
    P_Id int NOT NULL PRIMARY KEY,
```

```
    LastName varchar(255) NOT NULL,
```

```
    FirstName varchar(255),
```

```
    Address varchar(255),
```

```
    City varchar(255),
```

```
    CONSTRAINT chk_Person CHECK (P_Id>0 AND City='Sandnes')
```

```
)
```

INSERT INTO

INSERT INTO table_name

VALUES (value1,value2,value3,...);

OR

INSERT INTO table_name (column1,column2,column3,...)

VALUES (value1,value2,value3,...);

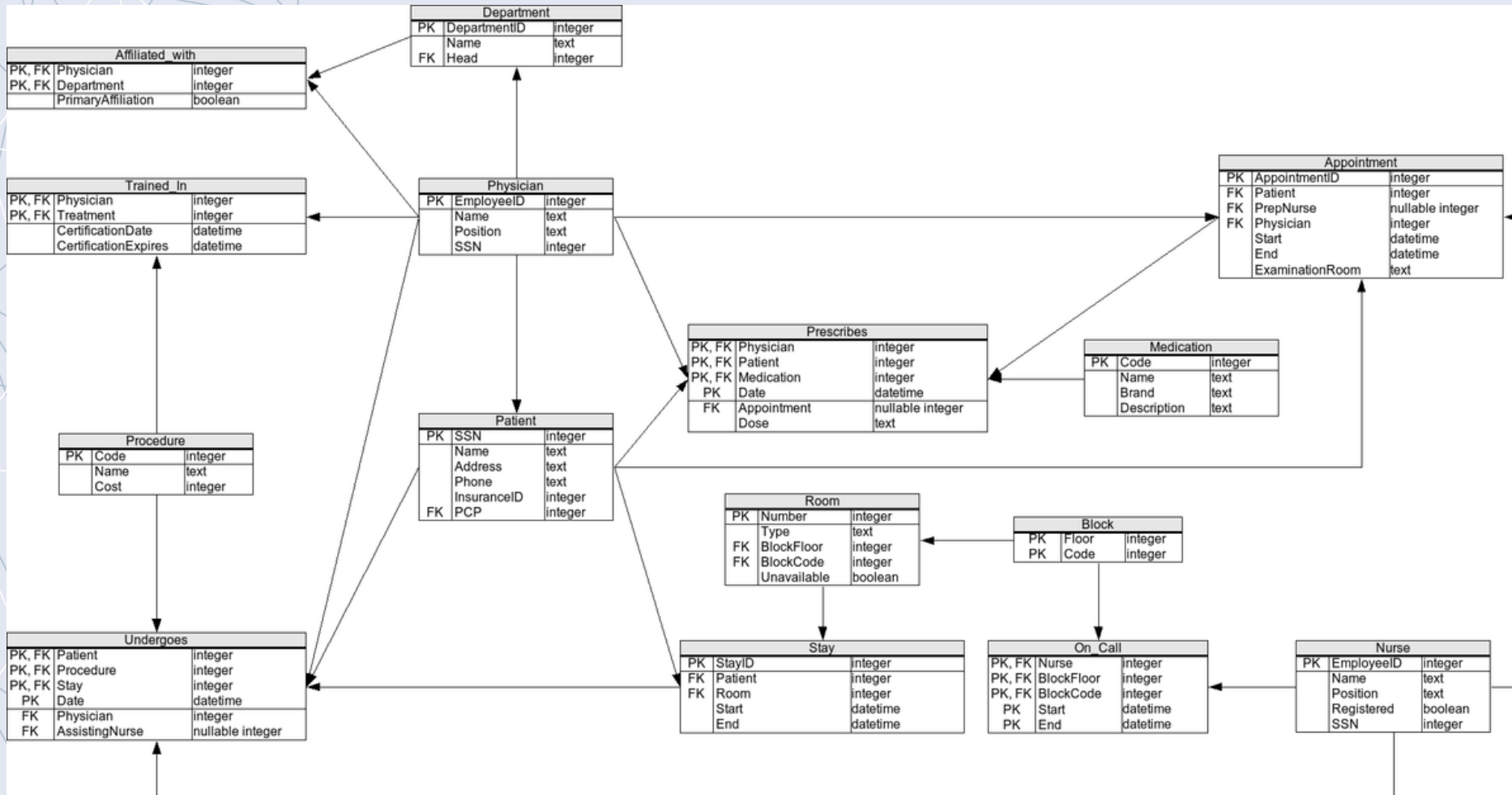
UPDATE

UPDATE table_name

SET column1=value1,column2=value2,...

WHERE some_column=some_value;

Hospital Database



SELECT

SELECT <*> | { [DISTINCT] <attribute_name> }

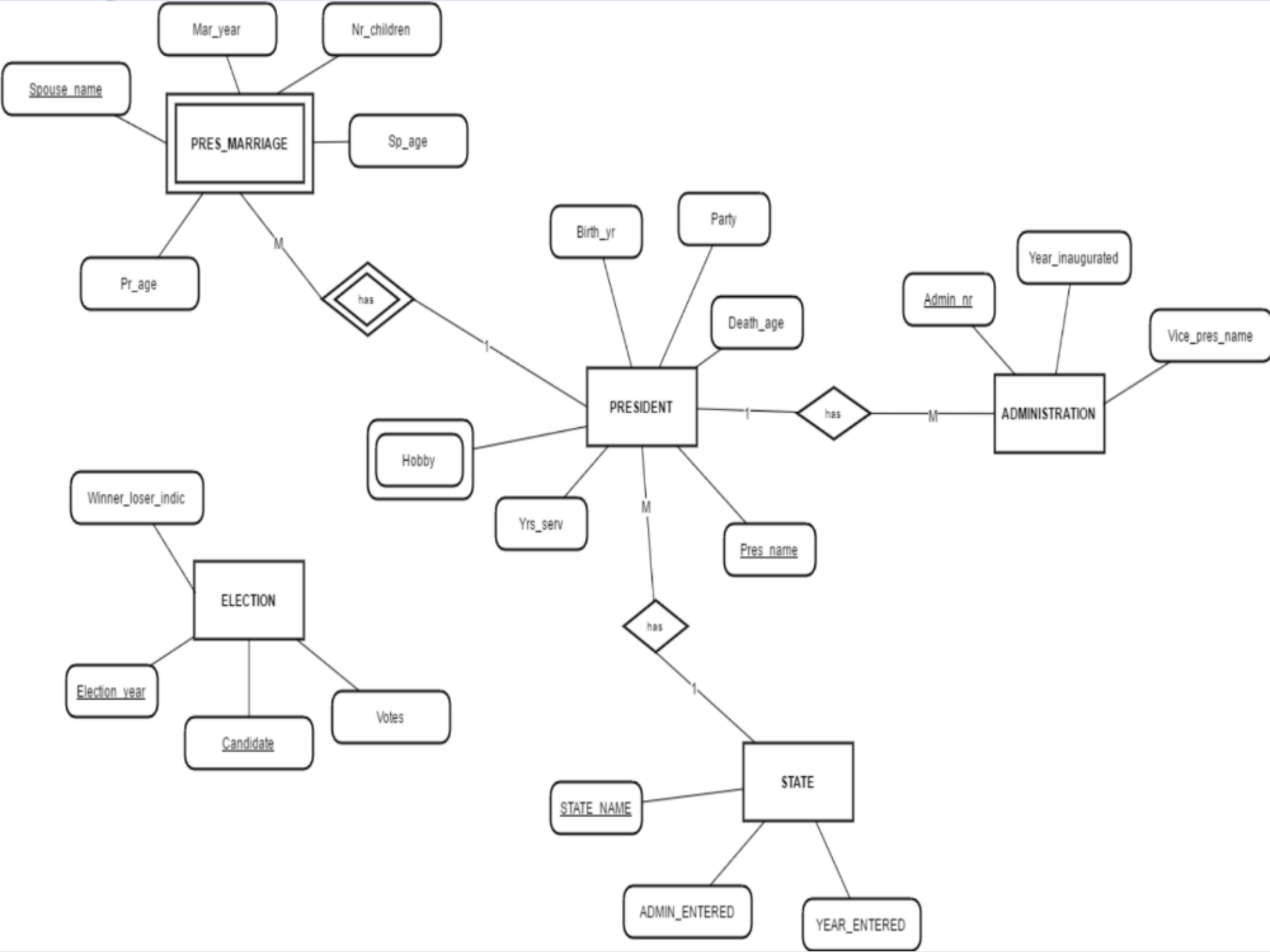
FROM <table_name>

[WHERE <condition>]

[GROUP BY <attribute_name>]

[HAVING <condition>]

[ORDER BY <attribute_name> [ASC | <DESC>]]



PRESIDENT Database

PRESIDENT

<u>Pres_name</u>	Birth_yr	Yrs_serv	Death_age	Party	State_born
------------------	----------	----------	-----------	-------	------------

ADMINISTRATION

<u>Admin_nr</u>	Pres_name	Year_inaugurated
-----------------	-----------	------------------

ADMIN_PR_VP

<u>Admin_nr</u>	<u>Pres_name</u>	<u>Vice_pres_name</u>
-----------------	------------------	-----------------------

PRESIDENT Database

STATE

<u>State_name</u>	Admin_entered	Year_entered
-------------------	---------------	--------------

ELECTION

<u>Election_year</u>	<u>Candidate</u>	Votes	Winner_loser_indic
----------------------	------------------	-------	--------------------

PRESIDENT Database

PRES_MARRIAGE

<u>Pres_name</u>	<u>Spouse_name</u>	Pr_age	Sp_age	Nr_children	Mar_year
------------------	--------------------	--------	--------	-------------	----------

PRES_HOBBY

<u>Pres_name</u>	<u>Hobby</u>
------------------	--------------

Example

- แสดงชื่อประธานาธิบดีจากตาราง PRESIDENT

```
SELECT PRES_NAME  
FROM PRESIDENT
```

Example

- แสดงชื่อประธานาธิบดีและงานอดิเรก

```
SELECT PRES_NAME, HOBBY  
FROM PRES_HOBBY
```

Example

- แสดงชื่อประธานาธิบดีและงานอดิเรก

```
SELECT *
```

```
FROM PRES_HOBBY
```

- ใช้เครื่องหมาย * แทนการระบุ Attribute ทุกตัวในตาราง

Example

- แสดงชื่อพรรคการเมืองจากราง PRESIDENT โดยไม่ซ้ำ

SELECT DISTINCT PARTY

FROM PRESIDENT

- การแสดงข้อมูลโดยไม่ให้ซ้ำเติม DISTINCT หลัง SELECT

Example

- แสดงชื่อประธานาธิบดี และพรรคการเมือง เรียงชื่อประธานาธิบดีตามตัวอักษร

```
SELECT PRES_NAME, PARTY  
FROM PRESIDENT  
ORDER BY PRES_NAME ASC
```

หรือ

```
SELECT PRES_NAME, PARTY  
FROM PRESIDENT  
ORDER BY PRES_NAME
```

Example

- แสดงชื่อประธานาธิบดี และปีเกิด เรียงผลตามปีเกิดจากมากไปน้อย

```
SELECT PRES_NAME, BIRTH_YR
```

```
FROM PRESIDENT
```

```
ORDER BY BIRTH_YR DESC
```

- ใส่ Attribute ที่ต้องการเรียงลำดับ หลัง ORDER BY
- การเรียงลำดับ มี 2 แบบ คือ
 - จากน้อยไปมาก ใช้ ASC
 - จากมากไปน้อย ใช้ DESC
- ถ้าไม่ได้ใส่ ASC หรือ DESC ไว้ จะเรียงแบบน้อยไปมาก

Example

- แสดงชื่อประธานาธิบดี และปีเกิด เรียงผลตามปีเกิดจากมากไปน้อย ถ้าปีเกิดเท่ากัน ให้เรียงลำดับตามชื่อประธานาธิบดี

```
SELECT PRES_NAME, BIRTH_YR
```

```
FROM PRESIDENT
```

```
ORDER BY BIRTH_YR DESC, PRES_NAME ASC
```


WHERE

- WHERE เป็นการกำหนดเงื่อนไขให้แสดงข้อมูลบางแถวที่สอดคล้องกับเงื่อนไข

- การใช้ WHERE

SELECT <*> | { [DISTINCT] <attribute> }

FROM <table_name>

WHERE <condition>

Comparison Operators

- การระบุเงื่อนไขที่ WHERE สามารถใช้ตัวดำเนินการเปรียบเทียบ (comparison operators)

= หมายถึง.....

^= หรือ <> หมายถึง.....

> หมายถึง.....

>= หมายถึง.....

< หมายถึง.....

<= หมายถึง.....

Arithmetic Operators

- ตัวดำเนินการเลขคณิต (arithmetic operators) ใช้เพื่อคำนวณทางคณิตศาสตร์
- ตัวดำเนินการเลขคณิตที่ใช้ $+$, $-$, $*$, $/$

Character Operators

- ใช้กับกรณีที่ต้องการข้อมูลที่มีตัวอักษรตามเงื่อนไข
- ใช้ LIKE และ NOT LIKE เป็นตัวเชื่อม
- ใช้ “_” (ขีดล่าง) แทน อักขระ 1 ตัว
- ใช้ “%” แทนอักขระตั้งแต่ 0 ตัวขึ้นไป

Logical Operator

- ตัวดำเนินการตรรกะ เป็นตัวดำเนินการที่ใช้เปรียบเทียบค่าสองค่า
 - จริง
 - เท็จ

เงื่อนไข	ความหมาย
AND	และ
OR	หรือ
[NOT] BETWEEN	มีค่าอยู่ระหว่าง
[NOT] IN	มีค่าอยู่ใน
IS [NOT] NULL	Null

Example

- แสดงชื่อประธานาธิบดีและคู่สมรสที่มีบุตรไม่เกิน 5 คน

```
SELECT PRES_NAME, SPOUSE_NAME  
FROM PRES_MARRIAGE  
WHERE NR_CHILDREN <= 5
```

Example

- จงแสดงชื่อประธานาธิบดีที่มีงานอดิเรกคือ ขี่ม้า

```
SELECT PRES_NAME
```

```
FROM PRES_HOBBY
```

```
WHERE HOBBY = "Riding"
```

Example

- แสดงปีที่เลือกตั้ง และชื่อผู้ชนะการเลือกตั้งในแต่ละปี โดยเรียงตามปีจากน้อยไปมาก

```
SELECT ELECTION_YEAR, CANDIDATE
```

```
FROM ELECTION
```

```
WHERE WINNER_LOSER_INDIC ="W"
```

```
ORDER BY ELECTION_YEAR ASC;
```


Example

- แสดงชื่อประธานาธิบดี รองประธานาธิบดี และ สมัยการเลือกตั้ง
ของประธานาธิบดีที่มีรองประธานาธิบดีเป็น Marshall T R

```
SELECT PRES_NAME, VICE_PRES_NAME, ADMIN_NR  
FROM ADMIN_PR_VP  
WHERE VICE_PRES_NAME = "Marshall T R"
```

Example

- แสดงชื่อประธานาธิบดีที่ขึ้นต้นด้วยตัวอักษร 'M'

```
SELECT PRES_NAME
```

```
FROM PRESIDENT
```

```
WHERE PRES_NAME LIKE 'M%'
```

Example

- จงแสดงชื่องานอดิเรกของประธานาธิบดีที่อักษรตัวที่สองของชื่อคือ 'a'

```
SELECT HOBBY
```

```
FROM PRES_HOBBY
```

```
WHERE PRES_NAME LIKE '_a%'
```

Example

- แสดงชื่อประธานาธิบดีที่ไม่มีตัวอักษร 'e' อยู่ในชื่อ

```
SELECT PRES_NAME
```

```
FROM PRESIDENT
```

```
WHERE PRES_NAME NOT LIKE '%e%'
```

Example

- แสดงชื่อประธานาธิบดีที่แต่งงานขณะอายุน้อยกว่า 30 ปี หรือคู่สมรสอายุไม่เกิน 25 ปี

```
SELECT PRES_NAME
```

```
FROM PRES_MARRIAGE
```

```
WHERE PR_AGE < 30 OR SP_AGE <= 25
```

Example

- แสดงชื่อประธานาธิบดีที่ไม่ได้ตกปลา หรือชกมวย หรือ อ่านหนังสือ เป็นงานอดิเรก

```
SELECT PRES_NAME
```

```
FROM PRES_HOBBY
```

```
WHERE HOBBY NOT IN ('Fishing', 'Boxing', 'Reading')
```

Example

- แสดงชื่อประธานาธิบดี ชื่อคู่สมรส และอายุคู่สมรส เฉพาะที่คู่สมรสอายุ 18 -25 ปี

```
SELECT PRES_NAME, SPOUSE_NAME, SP_AGE  
FROM PRES_MARRIAGE  
WHERE PR_AGE BETWEEN 18 AND 25
```

Example

- แสดงชื่อประธานาธิบดีที่ไม่มีตัวอักษร 'a' ทั้งตัวเล็กและตัวใหญ่
อยู่ในชื่อ

```
SELECT PRES_NAME
```

```
FROM PRESIDENT
```

```
WHERE PRES_NAME NOT LIKE '%a%' AND
```

```
PRES_NAME NOT LIKE '%A%'
```


Example

- แสดงชื่อประธานาธิบดีที่ขี่ม้าหรือล่าสัตว์

```
SELECT PRES_NAME
```

```
FROM PRES_HOBBY
```

```
WHERE HOBBY = 'Riding' OR HOBBY = 'Hunting'
```

หรือ

```
SELECT PRES_NAME
```

```
FROM PRES_HOBBY
```

```
WHERE HOBBY IN ('Riding', 'Hunting')
```

Example

- แสดงชื่อประธานาธิบดี ชื่อคู่สมรส จำนวนบุตร เฉพาะที่มีจำนวนบุตรไม่เกิน 3 คน และชื่อคู่สมรสที่มีตัวอักษร 'e' อยู่ในชื่อ

```
SELECT PRES_NAME, SPOUSE_NAME, NR_CHILDREN
```

```
FROM PRES_MARRIAGE
```

```
WHERE NR_CHILDREN <= 3 AND SPOUSE_NAME LIKE '%e%'
```

Example

- ใช้ “IS NULL” เมื่อต้องการค่าที่เป็น null
- ใช้ “IS NOT NULL” เมื่อไม่ต้องการค่าที่เป็น null
- จงแสดงชื่อประธานาธิบดีที่ถึงแก่อนิจกรรมแล้ว

```
SELECT PRES_NAME
```

```
FROM PRESIDENT
```

```
WHERE DEATH_AGE IS NOT NULL
```

Built-in Functions

- ฟังก์ชันในตัว มี 5 ฟังก์ชัน
 - AVG ค่าเฉลี่ย
 - SUM ผลรวม
 - MIN ค่าน้อยที่สุด
 - MAX ค่ามากที่สุด
 - COUNT นับจำนวนข้อมูล

Built-in Functions

ID	SCORE
1	4
2	7
3	6
4	7
5	3
6	9
7	6

- $\text{AVG}(\text{SCORE}) = 6$
- $\text{SUM}(\text{SCORE}) = 42$
- $\text{MIN}(\text{SCORE}) = 3$
- $\text{MAX}(\text{SCORE}) = 9$
- $\text{COUNT}(\text{SCORE}) = 7$

ข้อควรระวัง

- $\text{SUM}()$ ใช้หาผลรวม
- $\text{COUNT}()$ ใช้นับจำนวนข้อมูล

Built-in Functions

- เรียกฟังก์ชันในตัวต่อท้าย SELECT
- ไม่สามารถใช้ฟังก์ชันในตัวพร้อมกับการแสดง Attribute เช่น

```
SELECT PRES_NAME, COUNT(BIRTH_YR)  
FROM PRESIDENT
```

Example

- จงหาอายุเฉลี่ยของประธานาธิบดี

```
SELECT AVG(DEATH_AGE)  
FROM PRESIDENT
```

Example

- จงหาจำนวนปีที่ดำรงตำแหน่งมากที่สุดของประธานาธิบดี

```
SELECT MAX(YRS_SERV)  
FROM PRESIDENT
```


Example

- จงหาจำนวนลูกทั้งหมดของประธานาธิบดี 'Tyler J'

```
SELECT SUM(NR_CHILDREN)
```

```
FROM PRES_MARRIAGE
```

```
WHERE PRES_NAME = 'Tyler J'
```

Example

- จงหาจำนวนประธานาธิบดีที่สังกัดพรรค 'Federalist'

```
SELECT COUNT(*)
```

```
FROM PRESIDENT
```

```
WHERE PARTY = 'Federalist'
```

Example

- แสดงจำนวนเฉลี่ยของลูกประธานาธิบัติ ที่ประธานาธิบัติแต่งงานหลังปี 1900

```
SELECT AVG(NR_CHILDREN)
FROM PRES_MARRIAGE
WHERE MAR_YEAR > 1900
```

Questions & Answers